

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Томский государственный университет систем управления и радиоэлектроники»
(ТУСУР)

Кафедра автоматизированных систем управления

СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ В ЭВМ

Методические указания по лабораторным работам и организации
самостоятельной работы для студентов технических направлений подготовки
всех форм обучения

Томск-2022

Горитов А.Н.

Структуры и алгоритмы обработки данных в ЭВМ: методические указания по лабораторным работам и организации самостоятельной работы для студентов технических направлений подготовки всех форм обучения / А.Н. Горитов. – Томск: ТУСУР, 2022. – 17 с.

Одобрено на заседании кафедры автоматизированных систем управления.
Протокол № 11 от 23.11.2023 г.

© Томский государственный
университет систем управления и
радиоэлектроники, 2022
© Горитов А.Н. 2022

СОДЕРЖАНИЕ

1 Введение	4
2. Содержание дисциплины	5
3 Методические указания к проведению лабораторных работ.....	9
3.1 Общие положения	9
3.2 Лабораторные работы	9
4 Методические указания для организации самостоятельной работы.....	15
4.1 Общие положения	15
4.2 Проработка лекционного материала и подготовка к лабораторным работам.....	15
5 Рекомендуемые источники	16
5.1 Основные	16
5.2 Дополнительные	16

1 Введение

Целью изучения дисциплины «Структуры и алгоритмы обработки данных в ЭВМ» является изучение применяемых в программировании (и информатике) структур данных, их спецификации и реализации, а также алгоритмов обработки данных и анализ этих алгоритмов, взаимосвязь алгоритмов и структур.

В результате изучения дисциплины студент должен:

а) иметь представление об основных тенденциях в создании структур данных, методах оптимального использования памяти и времени для обработки структур данных и управления процессами обработки данных;

б) знать и использовать различные (динамические и статистические) структуры данных в соответствии с запросами алгоритмов;

в) создавать списковые и древообразные структуры и управлять организацией этих структур (изменение списков и деревьев посредством включения исключения, замены элементов структур) знать, использовать оптимальные методы поиска и сортировки данных;

г) знать и использовать основные алгоритмы решения классических задач информатики;

д) иметь представление о математических методах анализа алгоритмов; классификации алгоритмических задач по сложности, сводимости алгоритмических задач к известным задачам определенного класса сложности;

е) иметь опыт работы с алгоритмическими языками программирования.

Для успешного освоения дисциплины применяются различные образовательные технологии, которые обеспечивают достижение планируемых результатов обучения согласно основной образовательной программе, с учетом требований к объему занятий в интерактивной форме.

Интерактивные формы обучения, которые используются в данном курсе, включают: «Работа в команде» и «Поисковый метод».

Для контроля освоения компетенций используются следующие формы контроля: защита лабораторных работ, опрос по изучаемым разделам дисциплины, тесты.

Дисциплина «Структуры и алгоритмы обработки данных в ЭВМ» относится к вариативной части профессионального цикла ООП. Успешное овладение дисциплиной

предполагает предварительные знания таких дисциплин как «Математический анализ», «Дискретная математика», «Программирование», «Математическая логика и теория алгоритмов» в объеме, предусмотренном направлением 09.03.01 – «Информатика и вычислительная техника».

Знания и навыки, полученные при изучении этой дисциплины, используются при изучении дисциплин «Основы разработки программного обеспечения», «Теория языков программирования и методы трансляции», «Теория вычислительных процессов», «Базы данных».

2. Содержание дисциплины

Тема 1. Данные и ЭВМ

Предмет дисциплины и ее задачи. Связь с другими дисциплинами учебного плана направления и специальности.

Алгоритм. Вычислительная сложность алгоритма и ее оценка. Использование пределов для сравнения порядка роста двух функций. Основные классы эффективности.

Литература 1.

Тема 2. Фундаментальные структуры данных

Базовые типы данных, обрабатываемые командами ЭВМ. Представление чисел, символьных и логических данных, указателей в оперативной памяти. Понятие структуры данных. Классификация структур. Важнейшие операции над структурами.

Массивы, их представление в памяти. Информационный вектор. Строковые данные. Операции над строками.

Записи и структуры. Квалифицированные имена. Иерархия данных в записях. Записи с вариантами. Представление записей в памяти ЭВМ.

Множества. Операции над множествами. Представление в памяти.

Последовательный файл. Особенности файла как структуры данных. Основные действия над файлом. Файлы со сложной структурой. Прямой доступ.

Литература 1, 9.

Тема 3. Линейные динамические структуры

Структуры данных и алгоритмы. Стек, очередь и дек как линейные списки (последовательности) с ограниченными наборами операций (доступа). Стек, очередь и дек как абстрактные типы данных. Представление и реализация (непрерывная, ссылочная в

связанной памяти и на базе вектора).

Примеры алгоритмов, использующих стек, очередь, дек.

Связный список. Односвязные, двусвязные, кольцевые списки и операции над ними.

Представление и реализация (непрерывная, ссылочная в связанной памяти).

Литература 1, 9.

Тема 4. Древовидные структуры данных

Определение дерева, леса, бинарного дерева. Графическое и текстовое (скобочное) представление леса. Спецификация дерева, леса, бинарного дерева: базовые функции и аксиомы. Естественное соответствие бинарного дерева и леса.

Обходы бинарных деревьев: рекурсивные и не рекурсивные алгоритмы. Обходы дерева или леса.

Представления и реализации бинарных деревьев: ссылочная реализация в связанной памяти, ссылочная реализация ограниченного бинарного дерева на базе вектора.

Литература 1, 6, 7, 9.

Тема 5. Сортировка

Сортировка. Внутренняя сортировка. Стратегии внутренней сортировки. Сортировка выборкой: метод простого выбора, турнирная сортировка, пирамидальная сортировка, анализ сложности алгоритмов. Сортировка включением: метод простых вставок, метод вставки с убывающим шагом, анализ сложности алгоритмов. Обменные сортировки: сортировка пузырьком, быстрая сортировка, анализ сложности алгоритмов. Сортировка распределением: двоичная быстрая сортировка, цифровая распределяющая сортировка, блочная сортировка, сортировка подсчетом, анализ сложности алгоритмов. Сортировка слиянием.

Нижняя граница сложности задачи сортировки.

Внешняя сортировка. Простое слияние. Естественное слияние. Многофазная сортировка.

Литература 1, 4, 5, 9.

Тема 6. Исчерпывающий поиск

Исчерпывающий перебор. Примеры решения задач: задача коммивояжера, задача о рюкзаке, задача о назначениях.

Поиск с возвратом. Общий алгоритм. Пример. Другие способы программирования

поиска с возвратом: рекурсия и использование макросредств.

Метод ветвей и границ. Общая схема. Примеры применения метода решения задач: задача коммивояжера, задача о рюкзаке, задача о назначениях.

Динамическое программирование. Пример и общая идея. Вычисление чисел Фибоначчи. Восходящее и нисходящее динамическое программирование. Задача определения наиболее длинной общей подпоследовательности. Задача определения порядка умножения цепочки матриц

Литература 1, 3, 7, 5.

Тема 7. Быстрый поиск

Поиск и другие операции над таблицами. Последовательный и бинарный поиск. Бинарные деревья поиска. Случайные бинарные деревья поиска. Подсчет числа структурно различных бинарных деревьев с заданным числом узлов. Среднее время поиска в случайных деревьях.

Рандомизированные бинарные деревья поиска.

Сбалансированные по высоте бинарные деревья (АВЛ-деревья). Включение в АВЛ-дерево. Исключение из АВЛ-дерева. Оценка сложности в худшем случае: деревья Фибоначчи.

Красно-черные деревья (КЧ-деревья). Включение в КЧ-деревья. Исключение из КЧ-деревьев.

Реализация упорядоченных линейных списков на базе АВЛ-деревьев, КЧ-деревьев или рандомизированных деревьев. Операции поиска, вставки и удаления элементов; операции сцепления и расщепления списков.

2-3-деревья. Включение элемента в 2-3 дерево. Исключение элемента из 2-3 дерева. Поиск элемента в 2-3 дереве.

2-3-4-деревья. Включение элемента в 2-3-4 дерево. Исключение элемента из 2-3-4 дерева. Поиск элемента в 2-3-4 дереве.

Б-деревья. Включение элемента в Б-дерево. Исключение элемента из Б-дерева. Поиск элемента в Б-дереве.

Метод поиска с использованием функции расстановки (хеширование). Разрешение коллизий: метод внутренних и внешних цепочек, метод открытой адресации. Коэффициент загрузки, оценки сложности. Выбор функции расстановки.

Литература 1, 5, 7, 10.

Тема 8. Алгоритмы на графах

Графы: определения и примеры. Упорядоченный граф. Представления графов: матрица инцидентий, матрица смежности, список пар, списки смежности.

Поиск в графе: Основные методы обработки графов. Поиск в ширину. Поиск в глубину.

Связные компоненты: Определение компонент связности. Топологическая сортировка.

Двусвязность: Точки сочленения и их свойства. Алгоритм выделения компонент двусвязности графа.

Эйлеров путь в графе. Алгоритм построения Эйлерова пути.

Гамильтонов путь в графе. Нахождение Гамильтонова пути в графе с помощью алгоритма с возвратом.

Циклы: Фундаментальное множество циклов графа. Алгоритм отыскания фундаментального множества циклов в графе.

Остовные деревья графа: Связные компоненты. Построение и свойства остовных деревьев при поиске в глубину и в ширину.

Остовные деревья минимального веса: Минимальное остовное дерево. Алгоритм Прима. Алгоритм Крускала.

Кратчайшие пути в графе: Кратчайшие пути от фиксированной вершины. Алгоритм Форда-Беллмана. Алгоритм Дейкстры. Кратчайшие пути в бесконтурном графе.

Кратчайшие пути между всеми парами вершин: Матрица смежности, матрица достижимости и транзитивное замыкание отношения, алгоритм Уоршалла. Алгоритм Флойда-Уоршалла вычисления расстояний между всеми парами вершин, одновременное построение путей.

Литература 1, 2, 5, 7.

Тема 9. NP-полные и труднорешаемые задачи

Массовая и индивидуальная задачи. Сложность алгоритма и кодирование входных и выходных данных. Полиномиальные алгоритмы и класс P. Недетерминированные алгоритмы и класс NP.

Различные формы постановки задач комбинаторной оптимизации: оптимизационная, вычислительная, форма распознавания. Примеры.

Полиномиальная преобразуемость задач. NP-трудные и NP-полные задачи. Задача о выполнимости булева выражения, представленного в конъюнктивной нормальной форме.

Доказательство NP-полноты задачи о выполнимости.

Практическое решение NP-полных задач.

Литература 1, 5.

3 Методические указания к проведению лабораторных работ

3.1 Общие положения

Целью проведения лабораторных работ является формирование и развитие навыков программирования при реализации классических алгоритмов, изучаемых в этом курсе.

Основной формой проведения лабораторных работ является разработка алгоритма решения индивидуальной задачи и его программная реализация на языке Си. Процесс программной реализации включает в себя написание программы, отладку программы и тестирование программы.

К основным способам контроля формирования компетенций при выполнении лабораторных работ относятся индивидуальная защита выполненной работы, организация входного контроля уровня подготовки студентов по теоретическому материалу дисциплины, практическое применение которого осуществляется в ходе выполнения лабораторной работы.

Для получения максимальной оценки за лабораторную работу необходимо выполнить и защитить работу во время, отведенное для ее выполнения, согласно расписанию занятий. Допускается досрочное выполнение лабораторной работы по предварительной договоренности с преподавателем.

Выполнение всех лабораторных работ, предусмотренных рабочей программой дисциплины, является условием допуска к итоговому контролю изучения дисциплины – зачету или экзамену.

3.2 Лабораторные работы

По данной дисциплине выполняются следующие лабораторные работы:

1. Интервальные и перечислимые типы данных.
2. Операции над множествами.
3. Стеки, очереди
4. Связанные списки
5. Бинарные деревья
6. Сортировка
7. Фундаментальные алгоритмы на графах
8. Кратчайшие пути в графе

При подготовке к лабораторным работам необходимо внимательно разобрать материал, данный в лекции, а также обратиться к дополнительной литературе. Разобрать примеры, рассмотренные в лекциях и в литературе.

Лабораторная работа № 1.

Цель работы: закрепление навыков программирования с использованием интервальных и перечислимых типов данных.

Форма проведения: выполнение индивидуального задания.

Рекомендации по подготовке к занятию: перед проведением занятия необходимо изучить теоретический материал, изложенный в [1].

Порядок выполнения работы

1. Получить индивидуальный вариант.
2. Ответить на вопросы входного контроля
3. Составить алгоритм решения задачи индивидуального варианта.
4. Реализовать алгоритм на языке Си.
5. Выполнить тестирование полученного программного продукта
6. Защитить работу.

Контрольные вопросы по теме лабораторной работы

1. Что такое «перечислимый тип данных».
2. Операции, определенные для перечислимых типов данных.
3. Что такое «интервальный тип данных».
4. Операции, определенные для интервальных типов данных.

Лабораторная работа № 2.

Цель работы: закрепление навыков программирования с использованием множеств.

Форма проведения: выполнение индивидуального задания.

Рекомендации по подготовке к занятию: перед проведением занятия необходимо изучить теоретический материал, изложенный в [1].

Порядок выполнения работы

1. Получить индивидуальный вариант.
2. Ответить на вопросы входного контроля
3. Составить алгоритм решения задачи индивидуального варианта.
4. Реализовать алгоритм на языке Си.
5. Выполнить тестирование полученного программного продукта

6. Защитить работу.

Контрольные вопросы по теме лабораторной работы

1. Множественный тип данных.
2. Операции, выполняемые над множествами.

Лабораторная работа № 3

Цель работы: закрепление навыков программирования простейших структур на примере программирования стека и очереди.

Форма проведения: выполнение индивидуального задания.

Рекомендации по подготовке к занятию: перед проведением занятия необходимо изучить теоретический материал, изложенный в [1], а также ознакомиться с изучаемыми структурами данных в дополнительной литературе [2, 5, 11].

Порядок выполнения работы

1. Получить индивидуальный вариант.
2. Ответить на вопросы входного контроля
3. Составить алгоритм решения задачи индивидуального варианта.
4. Реализовать алгоритм на языке Си.
5. Выполнить тестирование полученного программного продукта
6. Защитить работу.

Контрольные вопросы по теме лабораторной работы

1. Сформулируйте принцип построения стека и очереди.
2. Запишите структуру языка Си для описания узла стека или очереди, если информационным полем будет число типа `int`.
3. Покажите на примере работу алгоритма добавления нового элемента в стек.
4. Покажите на примере работу алгоритма добавления нового элемента в очередь.
5. Покажите на примере работу алгоритма удаления элемента из стека.
6. Покажите на примере работу алгоритма удаления элемента из очереди.

Лабораторная работа № 4

Цель работы: закрепление навыков программирования динамических структур на примере программирования связанных списков.

Форма проведения: выполнение индивидуального задания.

Рекомендации по подготовке к занятию: перед проведением занятия необходимо изучить теоретический материал, изложенный в [1], а также ознакомиться с изучаемыми

структурами данных в дополнительной литературе [2, 5,].

Порядок выполнения работы

1. Получить индивидуальный вариант.
2. Ответить на вопросы входного контроля
3. Составить алгоритм решения задачи индивидуального варианта.
4. Реализовать алгоритм на языке Си.
5. Выполнить тестирование полученного программного продукта
6. Защитить работу.

Контрольные вопросы по теме лабораторной работы

1. Сформулируйте принцип построения связного списка.
2. Запишите структуру языка Си для описания узла связного списка, если информационным полем будет число типа `int`.
3. Покажите на примере работу алгоритма добавления нового элемента в связный список.
4. Покажите на примере работу алгоритма удаления элемента из связного списка.

Лабораторная работа № 5

Цель работы: закрепление навыков программирования иерархических структур на примере программирования деревьев двоичного поиска.

Форма проведения: выполнение индивидуального задания.

Рекомендации по подготовке к занятию: перед проведением занятия необходимо изучить теоретический материал, изложенный в [1] , а также ознакомиться с изучаемыми структурами данных в дополнительной литературе [2, 5, 11].

Порядок выполнения работы

1. Получить индивидуальный вариант.
2. Ответить на вопросы входного контроля
3. Составить алгоритм решения задачи индивидуального варианта.
4. Реализовать алгоритм на языке Си.
5. Выполнить тестирование полученного программного продукта
6. Защитить работу.

Контрольные вопросы по теме лабораторной работы

1. Сформулируйте принцип построения бинарного дерева поиска.
2. Запишите структуру языка Си для описания узла двоичного дерева, если информационным полем будет число типа `int`.

3. Покажите на примере работу алгоритма добавления нового элемента в дерево
4. Перечислите возможные способы обхода дерева.

Лабораторная работа № 6

Цель работы: ознакомиться и реализовать один из методов сортировки. Исследовать сложность реализованного алгоритма сортировки.

Форма проведения: выполнение индивидуального задания.

Рекомендации по подготовке к лабораторной работе: для выполнения лабораторной работы необходимо ознакомиться с алгоритмами сортировок, а также ознакомиться с изучаемыми структурами данных в дополнительной литературе [2, 5, 8].

Порядок выполнения работы

1. Получить индивидуальный вариант.
2. Составить алгоритм простой сортировки.
3. Реализовать алгоритм на языке Си, добавив в программу подсчет количества сравнений и перестановок, проведенных алгоритмом.
4. Найти среднее количество сравнений и перестановок, выполняемых программой для сортировки массивов из 100, 1000, 10000 элементов, результаты сохраните в текстовом файле.
5. Построить графики зависимости сложности алгоритма (количество сравнений + количество перестановок) от количества обрабатываемых данных.
6. Сделать выводы по работе.

Контрольные вопросы по теме лабораторной работы

1. Какими параметрами характеризуются алгоритмы сортировки?
2. В чем заключается сортировка методом вставки?
3. В чем заключается сортировка методом выбора?
4. В чем заключается сортировка методом обмена?

Лабораторная работа № 7

Цель работы: закрепление навыков программирования сложных структур на примере программирования фундаментальных алгоритмов на графах.

Форма проведения: выполнение индивидуального задания.

Рекомендации по подготовке к занятию: перед проведением занятия необходимо изучить теоретический материал, изложенный в [1], а также ознакомиться с изучаемыми структурами данных в дополнительной литературе [2, 6, 6, 8, 9, 10].

Порядок выполнения работы

1. Получить индивидуальный вариант.
2. Ответить на вопросы входного контроля
3. Составить алгоритм решения задачи индивидуального варианта.
4. Реализовать алгоритм на языке Си.
5. Выполнить тестирование полученного программного продукта
6. Защитить работу.

Контрольные вопросы по теме лабораторной работы

1. Способы представления графа?
2. Что такое "связный граф"?
3. Что такое "путь" в графе?
4. Что такое "цикл" в графе?

Лабораторная работа № 8

Цель работы: закрепление навыков программирования сложных структур на примере программирования алгоритмов нахождения кратчайших путей в графе.

Форма проведения: выполнение индивидуального задания.

Рекомендации по подготовке к занятию: перед проведением занятия необходимо изучить теоретический материал, изложенный в [1], а также ознакомиться с изучаемыми структурами данных в дополнительной литературе [2, 6, 6, 8, 9, 10].

Порядок выполнения работы

1. Получить индивидуальный вариант.
2. Ответить на вопросы входного контроля
3. Составить алгоритм решения задачи индивидуального варианта.
4. Реализовать алгоритм на языке Си.
5. Выполнить тестирование полученного программного продукта
6. Защитить работу.

Контрольные вопросы по теме лабораторной работы

1. Что такое "Минимальное покрывающее дерево"?
2. Алгоритм Прима.
3. Алгоритм Крускала.
4. Нахождение кратчайших путей в графе.

4 Методические указания для организации самостоятельной работы

4.1 Общие положения

Самостоятельная работа является важной составляющей в изучении дисциплины и состоит из следующих видов деятельности: проработка лекционного материала для подготовки к тестированию и контрольным работам, подготовка к лабораторным работам и практическим заданиям, выполнение контрольных работ, самостоятельное изучение тем курса. Самостоятельная работа над теоретическим материалом направлена на систематизацию и закрепление знаний, полученных на лекционных занятиях и на получение новых знаний по дисциплине, путем самостоятельного изучения тем.

Самостоятельная работа по подготовке к лабораторным работам направлена на изучение методического и теоретического материала по теме лабораторной работы.

Выполнение контрольных работ - полностью самостоятельная работа, направленная на получение навыков самостоятельного составления алгоритмов, реализацию программ, их дальнейшей отладки и тестирования.

4.2 Проработка лекционного материала и подготовка к лабораторным работам

Проработка лекционного курса является одной из важных активных форм самостоятельной работы. Этот вид самостоятельной работы может быть организован следующим образом:

- прочитайте конспект лекции, согласуя Ваши записи с информацией на слайдах лекции;
- попробуйте выполнить самостоятельно примеры программ, разобранных на лекции;
- если в лекции рассматривался какой-либо алгоритм, попытайтесь выполнить этот алгоритм на тестовых данных без использования компьютерной программы; такой способ проработки материалов лекции покажет, правильно ли Вы поняли идею алгоритма;
- изучите дополнительные учебные материалы, рекомендованные преподавателем;
- попытайтесь ответить на контрольные вопросы, которыми, как правило, заканчиваются разделы учебных пособий или учебников;
- если после выполненной работы Вы считаете, что материал освоен не полностью, сформулируйте вопросы и задайте их преподавателю.

Методические указания к ведению конспектов лекций.

Лекции по дисциплине проводятся с использованием слайдов. Но это не означает, что лекцию можно просто слушать. Ведение конспектов значительно повышает качество

последующей проработки лекционного материала. В силу специфики дисциплины на слайдах лекций много алгоритмов, кодов программ, примеров демонстрации работы изучаемых алгоритмов. Но этот материал может быть бесполезен, если Вы не делаете записи в течение лекции, потому что большинстве случаев, комментарии по представленным на слайдах примерам, лектор выполняет в устной форме.

Самостоятельная работа по подготовке к лабораторным работам.

Самостоятельная работа по подготовке к лабораторным работам по дисциплине состоит в изучении методических материалов по темам соответствующих видов аудиторных занятий.

Рекомендуется перед выполнением лабораторной работы изучить лекционный и методический материал по теме занятия, ознакомиться с алгоритмами, реализацию которых необходимо выполнить во время проведения занятия. Обратите особое внимание на порядок выполнения работы. Поскольку конечным результатом всех лабораторных работ является компьютерная программа, самостоятельно разработайте структурную схему будущей программы, выполните заготовку проекта, подготовьте самостоятельно тестовые данные.

Перед проведением практического занятия подготовьте ответы на вопросы, вынесенные для проведения входного контроля уровня знаний по теме занятия. Если при подготовке к занятию остались нерешенные вопросы, обратитесь за консультацией к преподавателю.

5 Рекомендуемые источники

5.1 Основные

1. Гагарина Л.Г., Колдаев В.Д. Алгоритмы и структуры данных: Учебное пособие. – М.: Финансы и статистика; ИНФРА-М, 2009. – 304 с. (60 экз.)

5.2 Дополнительные

2. Горитов А.Н. Основы структур и алгоритмов обработки данных: Учебное пособие. – Томск: ТУСУР, 2007. – 229 с. (50 экз.)

3. Окулов С. М. Программирование в алгоритмах. – 2-е изд., доп. – М.: БИНОМ. Лаборатория знаний, 2006. – 384 с. (30 экз.)

4. Иванов Б.Н. Дискретная математика. Алгоритмы и программы: Учебное пособие. – М.: Лаборатория базовых знаний, 2003. – 288 с. (50 экз.)

5. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989. – 360 с. (50 экз.)
6. Макконелл Дж. Основы современных алгоритмов. 2-е дополненное издание. – Москва: Техносфера, 2004. – 368 с. (13 экз.)
7. Ускова О.Ф. и др. Программирование алгоритмов обработки данных: Учебное пособие. – СПб: БХВ-Петербург, 2003. – 188 с. (19 экз.)
8. Андерсон Д.А. Дискретная математика и комбинаторика. – М.: Издательский дом "Вильямс", 2004. – 960 с. (10 экз.)
9. Новиков Ф.А. Дискретная математика для программистов: Учебник для вузов. – СПб: Питер, 2002. – 302 с. (19 экз.)
10. Липский В. Комбинаторика для программистов. – М.: Мир, 1989. – 213 с. (9 экз.)
11. Кнут Д. Искусство программирования для ЭВМ. Том 1: Основные алгоритмы. – М.: Мир, 1976. – 736 с. (3-е изд.: Уч. пос. – М.: Издательский дом «Вильямс», 2000. – 720 с.) (5 экз.)
12. Кнут Д. Искусство программирования для ЭВМ. Том 3: Сортировка и поиск. – М.: Мир, 1978. – 846 с. (2-е изд.: Уч. пос. – М.: Издательский дом «Вильямс», 2000. – 832 с.) (15 экз.)
13. Костин А.Е., Шаньгин В.Ф. Организация и обработка структур данных в вычислительных системах. – М.: Высшая школа, 1987. – 248 с. (9 экз.)
14. Стоун Г.С., Сиворек Д.П. Введение в организацию ЭВМ и структуры данных. – М.: Машиностроение, 1980. – 320 с. (10 экз.)